scientific reports

Check for updates

OPEN

Coupling a recurrent neural network to SPAD TCSPC systems for real-time fluorescence lifetime imaging

Yang Lin, Paul Mos, Andrei Ardelean, Claudio Bruschini & Edoardo Charbon ✉

Fluorescence lifetime imaging (FLI) has been receiving increased attention in recent years as a powerful diagnostic technique in biological and medical research. However, existing FLI systems often suffer from a tradeoff between processing speed, accuracy, and robustness. Inspired by the concept of Edge Artificial Intelligence (Edge AI), we propose a robust approach that enables fast FLI with no degradation of accuracy. This approach couples a recurrent neural network (RNN), which is trained to estimate the fluorescence lifetime directly from raw timestamps without building histograms, to SPAD TCSPC systems, thereby drastically reducing transfer data volumes and hardware resource utilization, and enabling real-time FLI acquisition. We train two variants of the RNN on a synthetic dataset and compare the results to those obtained using center-of-mass method (CMM) and least squares fitting (LS fitting). Results demonstrate that two RNN variants, gated recurrent unit (GRU) and long short-term memory (LSTM), are comparable to CMM and LS fitting in terms of accuracy, while outperforming them in the presence of background noise by a large margin. To explore the ultimate limits of the approach, we derive the Cramer-Rao lower bound of the measurement, showing that RNN yields lifetime estimations with near-optimal precision. To demonstrate real-time operation, we build a FLI microscope based on an existing SPAD TCSPC system comprising a 32x32 SPAD sensor named Piccolo. Four quantized GRU cores, capable of processing up to 4 million photons per second, are deployed on the Xilinx Kintex-7 FPGA that controls the Piccolo. Powered by the GRU, the FLI setup can retrieve real-time fluorescence lifetime images at up to 10 frames per second. The proposed FLI system is promising and ideally suited for biomedical applications, including biological imaging, biomedical diagnostics, and fluorescence-assisted surgery, etc.

Fluorescence lifetime imaging (FLI) is an imaging technique for the characterization of molecules based on decay time from the excited state to the ground state[1]. Compared with fluorescence intensity imaging, FLI is insensitive to excitation intensity fluctuations, variable probe concentration, and limited photobleaching. Besides, through the appropriate use of targeted fluorophores, FLI is able to quantitatively measure the parameters of the microenvironment around fluorescent molecules, such as pH, viscosity, and ion concentrations[2,3]. With these advantages, FLI has wide applications in the biological sciences, for example to monitor protein-protein interactions[4], and plays an increasing role in medical and clinical settings such as visualization of tumor margins[5], cancerous tissue detection[1,6], and computer-assisted robotic surgery[7,8].

Time-correlated single-photon counting (TCSPC) is popular among FLI systems due to its superiority over other techniques in terms of time resolution, dynamic range, and robustness. In TCSPC, one records the arrival time of individual photons emitted by molecules upon photoexcitation[9–11]. After repeated measurements, one can construct a histogram of photon arrivals, which closely matches the true response of molecules, thus enabling the extraction of FLI, as shown Fig. 1. The instrumentation of a typical TCPSC FLI system features a confocal setup, including a single-photon detector, a dedicated TCSPC module for time tagging, and a PC for lifetime estimation[9,12]. Such systems are mostly unsuitable for increasing clinical applications such as non-invasive monitoring, where a miniaturized and fast FLI system is desired[13]. Additionally, the substantial volume of data produced by TCSPC imposes a significant load on data transfer, storage, and processing. A powerful PC, sometimes equipped with dedicated GPUs, is required to acquire and process TCSPC data. TCSPC requires photodetectors

Advanced Quantum Architecture Laboratory, École polytechnique fédérale de Lausanne, Neuchâtel 2002, Switzerland. ✉email: edoardo.charbon@epfl.ch

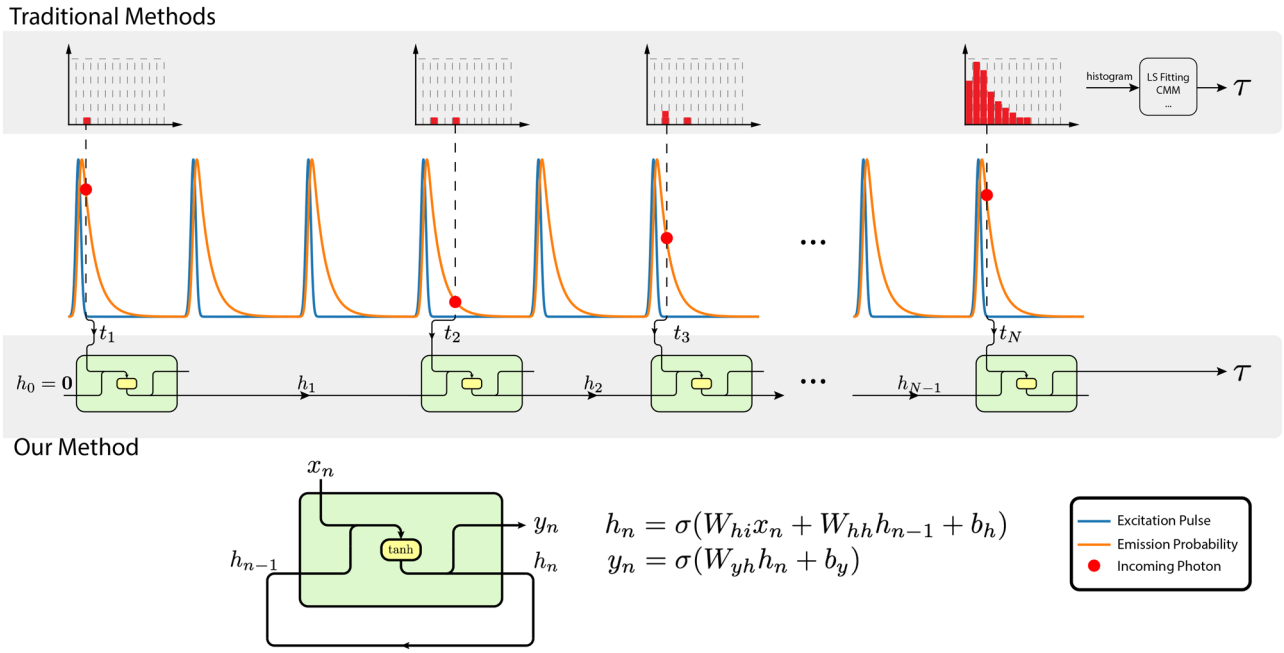


Figure 1. In a traditional TCSPC FLI system, the sample is excited by a laser repeatedly, and the emission photons are detected and time-tagged. A histogram is gradually built on these timestamps, from which the lifetime can be extracted after the acquisition is completed. In our proposed system, upon the receiving of a photon, the timestamp is fed into the RNN immediately. The RNN updates the hidden state accordingly and idles for the next photon. The schematic and formula of simple RNN are shown here[37]. At timestep n, the RNN takes the current information x_n and the past information h_{n-1} as input, then updates the memory to the current information h_n and gives out a prediction y_n. W_{hi}, W_{hh}, W_{hi}, b_h, and b_y are the weights and biases to be learned from training. $\sigma(\cdot)$ is the non-linear activation function, which is usually the hyperbolic tangent (tanh).

with picosecond time resolution and single-photon detection capability. In the last decade, single-photon avalanche diodes (SPADs) have been used successfully in TCSPC systems and, with the advent of CMOS SPADs, the expansion of these detectors into high-resolution image sensors for widefield imaging has been accomplished[14]. Several reviews of the use of SPADs in biophotonics have recently appeared[15–17].

Least squares (LS) fitting and maximum likelihood estimation (MLE) are widely used for fluorescence lifetime estimation[18–20]. These two methods rely on iterations to achieve high accuracy, but they are time-consuming and computationally expensive. Various non-fitting methods have been proposed to tackle these problems but often compromise on other specifications, among which the Center-of-Mass method (CMM) is a typical one. CMM is a simple, fast, and photon-efficient alternative, which has been already applied in some real-time FLI systems[21–23]. However, it is very sensitive to background noise, and the estimation is biased due to the use of truncated histograms[24].

Neural networks provide a new path to fast fluorescence lifetime extraction[25]. The first neural network-based model for fluorescence lifetime estimation was proposed in 2016, where higher accuracy and faster processing than LS fitting were reported[26]. Since then, several neural network architectures, including fully connected neural network (FCNN), convolutional neural network (CNN), and generative adversarial network (GAN) solutions have been explored for this end[27–31]. These techniques showed the ability to resolve multi-exponential decays and achieve accurate and fast estimation even in low photon-count scenarios. Apart from fluorescence lifetime determination, these neural networks can extract high-level features and can be integrated into a large-scale neural network for end-to-end lifetime image analysis such as cancerous tissue margin detection[32] and microglia detection[33]. To date, neural networks have primarily been considered a non-fitting alternative for lifetime estimation at the software level. Few studies have delved into their potential as a near-sensor solution for rapid, low-latency processing. While some research has explored on-FPGA implementations of these neural networks, there hasn't been the actual development of a practical real-time FLI system based on this approach as of yet[34–36]. Moreover, these existing efforts have remained rooted in using histograms as input, without fully exploring the promising prospect of directly employing timestamps as neural network input, which would bring the network even closer to the sensor.

Herein, we introduce an innovative approach that embraces the concept of Edge Artificial Intelligence (Edge AI), by integrating a recurrent neural network (RNN) into SPAD TCSPC systems for real-time FLI. The operational concept is visually explained in Fig. 1. Unlike conventional deep learning methods that rely on histograms as input, which are only available once the data acquisition is complete, RNNs eliminate histogramming and process raw timestamps on the fly in an event-driven way. This approach enables the continuous and incremental updating of lifetime estimations with each incoming photon, so the lifetime can be read out during or right after the acquisition. This novel methodology eliminates the necessity to store or transfer timestamp data or histograms, substantially reducing the burden on hardware memory and data transfer requirements. By

taking timestamps as input and producing lifetimes as output, the neural network's size is significantly reduced compared to existing models, with the number of parameters reduced to mere hundreds. Consequently, this approach minimizes hardware resource demands and latency, making it more feasible for implementation on hardware platforms.

In this work, we simulate the fluorescence process and create synthetic datasets on the timestamp level. On these datasets, variants of RNNs for lifetime estimation are trained and evaluated. These RNNs excel in providing fast, accurate, and robust lifetime estimation, showing superior performance in noisy scenes. To further bring the RNNs to the hardware, we quantize the RNNs and implement them on the FPGA with high-level synthesis (HLS). To showcase their real-time functionality, we build a FLI system based on Piccolo, a 32×32 SPAD sensor previously developed at EPFL[38]. Enabled by the small-scale architecture, a Gated Recurrent Unit (GRU), one of the RNN variants, is trained, quantized, and implemented on the same FPGA that controls and communicates with Piccolo, optimizing data throughput and minimizing the latency. From photon detection to lifetime estimation, the whole system is integrated into a compact, portable device, which achieves much lower data bandwidth consumption and much lower power consumption. The miniaturized real-time FLI system can benefit various biomedical applications such as fluorescence-assisted surgery and diagnosis. With the flexibility to retrain neural networks, the same system can be easily reused for other very different applications such as near-infrared optical tomography (NIROT)[39].

Results

The proposed system comprises a SPAD image sensor with timestamping capability coupled to an FPGA for implementation of neural networks in situ. In this section, we describe the utilized RNN, its training, and the achieved results. We also describe the upper bounds on accuracy that were derived to contextualize the results obtained with the RNNs.

RNNs trained on synthetic datasets and performance

We train and evaluate RNNs on synthetic datasets. Three RNN variants, namely simple RNN, gated recurrent unit (GRU)[40], and long short-term memory (LSTM)[41], are adopted. These RNNs are constructed with 8, 16, and 32 hidden units, respectively. LS fitting and CMM are also benchmarked. The metrics used for evaluation are the root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). A common range of lifetime is covered here, which is from 0.2 to 5 ns, and we assume a laser repetition frequency of 20 MHz.

We start with a simple situation in which background noise is absent. All the tests are run on a PC with 32-bit floating point (32FP) precision. The results are presented in Table 1. One can observe that CMM achieves the lowest error in MAE and MAPE, GRU-32 achieves the lowest RMSE error, and GRU-32 and LSTM-32 have very similar performance to CMM. The performance of CMM itself is understandable. In this case, background noise is not considered, and the repetition period is 10 times the longest lifetime. Under these conditions, CMM is very close to the maximum likelihood estimator of the lifetime. When comparing Simple RNN, GRU, and LSTM, one can observe that GRU outperforms LSTM by a small margin, and both of them perform much better than Simple RNN. As we can see with the decrease in model size, errors increase accordingly.

Background noise is often inevitable during fluorescence lifetime imaging, especially in diagnostic and clinical setups where the interruption to existing workflows is supposed to be minimized[13]. In our FLI system, it is estimated that at least 1% of the collected timestamps are from background noise. Therefore, we study the performance of each method under varying background noise levels. For simplicity, only LSTM-32 is used to compare with benchmarks. LSTM-32 is trained on a synthetic dataset, where 0 to 10% uniform background noise is added

Model	RMSE	MAE	MAPE
LS fitting	0.1642	0.1201	0.0553
CMM	**0.0915**	**0.0642**	**0.0250**
Simple RNN-8	0.2516	0.1979	0.0969
Simple RNN-16	0.2396	0.1798	0.0771
Simple RNN-32	0.1877	0.1415	0.0659
GRU-8	0.0957	0.0695	0.0297
GRU-16	0.0928	0.0666	0.0274
GRU-32	**0.0908**	**0.0647**	**0.0261**
LSTM-8	0.0981	0.0720	0.0423
LSTM-16	0.0928	0.0669	0.0277
LSTM-32	0.0916	0.0656	0.0267

Table 1. RNN models are trained and tested on a synthetic dataset, where the fluorescence decay model is mono-exponential, lifetime ranges from 0.2 and 5 ns, laser repetition frequency is 20 MHz, and background noise is not considered. Their performance is benchmarked against the Least Squares (LS) fitting and the Center-of-Mass method (CMM). RMSE: root mean squared error, MAE: mean absolute error, MAPE: mean absolute percentage error. Significant values are in bold.

to the samples randomly. Here we also illustrate the result of CMM with background noise subtraction, assuming that the number of photons from background noise is known, though it is often not the case in real-time FLI systems. Two synthetic datasets are built for evaluation, where the background noise ratios are 1% (SNR=20dB) and 5% (SNR=12.8dB), respectively. The results are presented in Table 2. We can see that LSTM-32 outperforms other methods in all metrics and scenarios. Combined with Table 1, one can observe that errors increase when the background noise increases for all the methods. However, LSTM and LS fitting are more robust to background noise, while CMM is extremely sensitive to it. This finding is in agreement with previous studies[1,42].

Cramer-rao lower bound analysis

To compare the performance with the theoretical optima, the Cramer-Rao lower bound (CRLB) is calculated of the accuracy of the lifetime estimate with an open source software[42], given the setting parameters. The variance of the lifetime estimation methods is calculated from Monte Carlo experiments. As for CMM and RNN, 3000 samples are used; as for the least squares method, 1000 samples are used to reduce running time.

The relationship between lifetime and the relative standard deviation of the different estimators is shown in Fig. 2a, where the photon count is 1024. One can observe that the variance of CMM and LSTM-32 almost reaches the CRLB, which suggests that CMM and LSTM-32 are near-optimal estimators. Considering that the laser repetition period is much longer than the lifetime and that background noise is not included, it is understandable that CMM reaches the CRLB, since it is approximately a maximum likelihood estimator. LS fitting performs worse than CMM and LSTM-32, which is likely due to the underlying assumption of Gaussian errors.

The relationship between the number of photons and the relative standard deviation of the different estimators is shown in Fig. 2b, where the lifetime is set at 2.5 ns. Similar to Fig. 2a, the relative standard deviations of CMM and LSTM-32 almost reach the CRLB, while the least square fitting performs worse. This result suggests that CMM and LSTM-32 are efficient estimators over different photon inputs, achieving excellent photon efficiency. They only need less than half of the data to obtain similar results as LS fitting.

We also analyze the CRLB with background noise. The results are shown in Fig. 2. Comparing Fig. 2c and Fig. 2e with Fig. 2a, we can see the CRLB marginally rises in the presence of background noise. The relative standard deviation of LS fitting stays almost unchanged, and that of LSTM-32 increases slightly but is still much better than LS fitting. As for CMM, one can see that the relative standard deviation increases dramatically at shorter lifetimes, which suggests that CMM is very sensitive to background noise for short lifetimes. By comparing Fig. 2d and Fig. 2f with Fig. 2b, we find that the relative standard deviation does not vary with 1% background noise. With 5% background noise, however, CMM shows a clear degradation of performance, its relative standard deviation getting close to the one of LS fitting.

Performance on experimental dataset

To verify the performance of RNNs, which are purely trained on synthetic datasets, on real-world data, the RNNs are tested on experimental data along with CMM and LS fitting as benchmarks. We prepare a fluorescence lifetime-encoded microbeads sample and acquire the TCSPC data with a commercial confocal FLIM setup (See Sect. "Methods–Dataset" for details). It is estimated that the background noise is below 1%. The LSTM-32 is trained on the dataset with varying background noise levels. For each data point, represented as a series of timestamps, a background noise ratio is sampled uniformly between 0% and 10%. This ratio determines the proportion of timestamps that originate from background noise. The corresponding results are shown in Fig. 3.

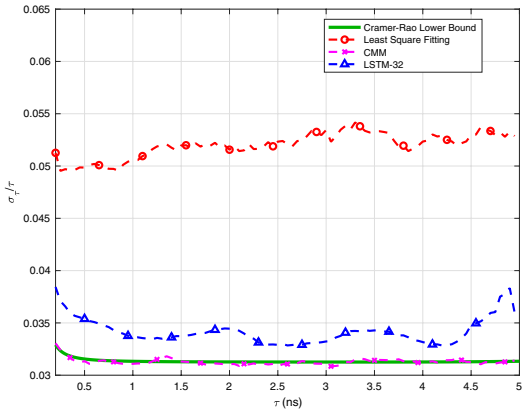
The histograms of the three samples share a similar shape. As for LS fitting, an instrument response function (IRF) is estimated from histograms of all pixels and then shared among them, which accounts for its good performance here. The result of CMM has a 2 ns bias, which is corrected by the estimated IRF. It is worth noting that for the first peak in the histogram, LSTM shows a sharper Gaussian shape, which confirms LSTM's good performance under low fluorescence intensity and short lifetime.

Real-time FLIM setup with on-FPGA RNN

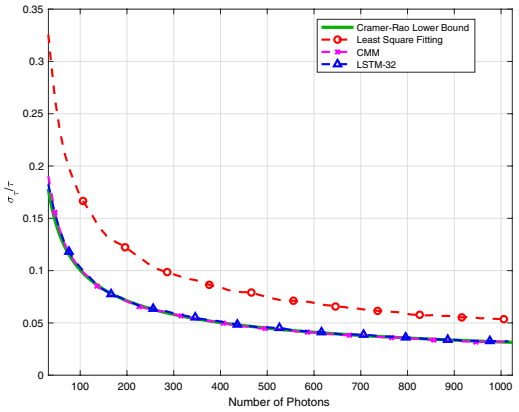
We further built a real-time FLIM system by utilizing a SPAD array sensor with on-chip time-to-digital converters (TDCs) and deploying the aforementioned RNNs on FPGA for near-sensor processing. The schematic of our setup is shown in Fig. 4. The 32×32 Piccolo SPAD sensor developed at EPFL[38,43] is utilized, on which 128 TDCs offer 50 ps temporal resolution. The sensor is controlled by a Kintex-7 FPGA, where four GRU-12 cores are implemented for lifetime estimation. The GRU-12 is trained on the PC, whose weights are quantized to 16-bit

	1% background noise			5% background noise		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LS fitting	0.1678	0.1226	0.0562	0.1883	0.1368	0.0609
CMM	0.2367	0.2168	0.1577	1.0742	1.0635	0.7799
CMM*	0.1099	0.0839	0.0456	0.2476	0.2128	0.1444
LSTM-32	**0.1019**	**0.0733**	**0.0304**	**0.1097**	**0.0784**	**0.0323**

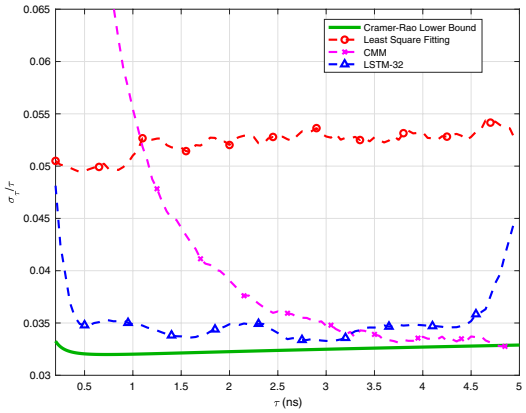
Table 2. Performance of LS fitting, CMM, CMM with background subtraction, and LSTM-32 in the presence of 1% and 5% background noise. *CMM with background noise subtraction. LSTM-32 is trained on a dataset including 0% to 10% random background noise for generalization in different scenarios. Significant values are in bold.



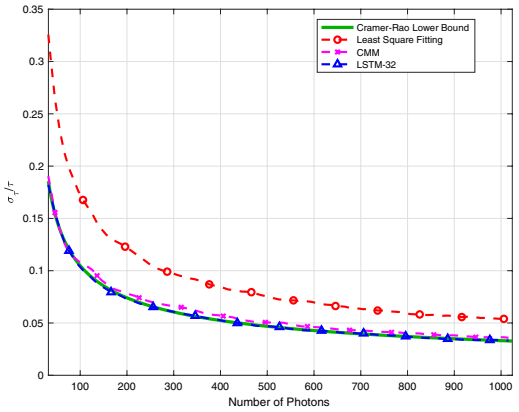
(a) Relative standard deviation of different estimation methods and CRLB as a function of lifetime for a fixed number of detected photons (1024). No background noise.



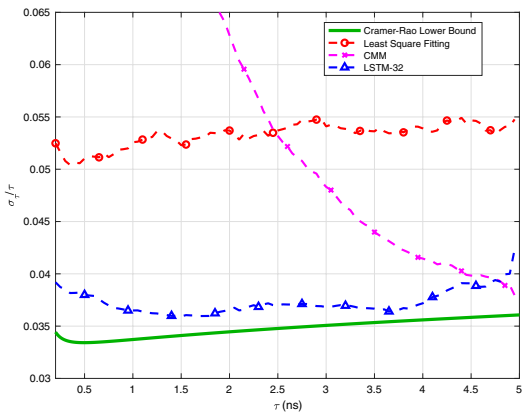
(b) Relative standard deviation of different methods and CRLB as a function of the number of detected photons for a fixed lifetime of 2.5 ns. No background noise.



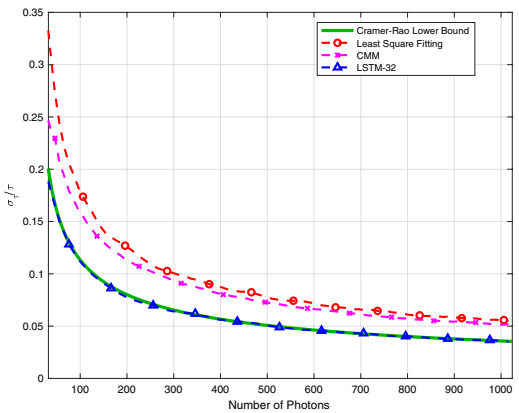
(c) Relative standard deviation of different estimation methods and CRLB as a function of lifetime for a fixed number of detected photons (1024). 1% background noise.



(d) Relative standard deviation of different methods and CRLB as a function of the number of detected photons for a fixed lifetime of 2.5 ns. 1% background noise.



(e) Relative standard deviation of different estimation methods and CRLB as a function of lifetime for a fixed number of detected photons (1024). 5% background noise.



(f) Relative standard deviation of different methods and CRLB as a function of the number of detected photons for a fixed lifetime of 2.5 ns. 5% background noise.

Figure 2. Cramer-Rao lower bound analysis when including 0%, 1%, and 5% background noise levels.

fixed point numbers and finetuned with quantization-aware training. The quantized GRU has an acceptable drop in accuracy and still outperforms benchmarks (i.e. LS fitting and CMM) by a large margin. The four GRU cores are able to process up to 4 million photons per second. While the data transfer rate to the PC is 20Mb/s for histogram mode and 80Mb/s for raw mode, it reduces to only 240kb/s when applying the proposed RNN-based lifetime estimation method.

We prepare a sample containing fluorescent beads with a lifetime of 5.5 ns. The sample is measured by our system in real-time at 5 frames per second. During the imaging, we move the sample plate forward to observe

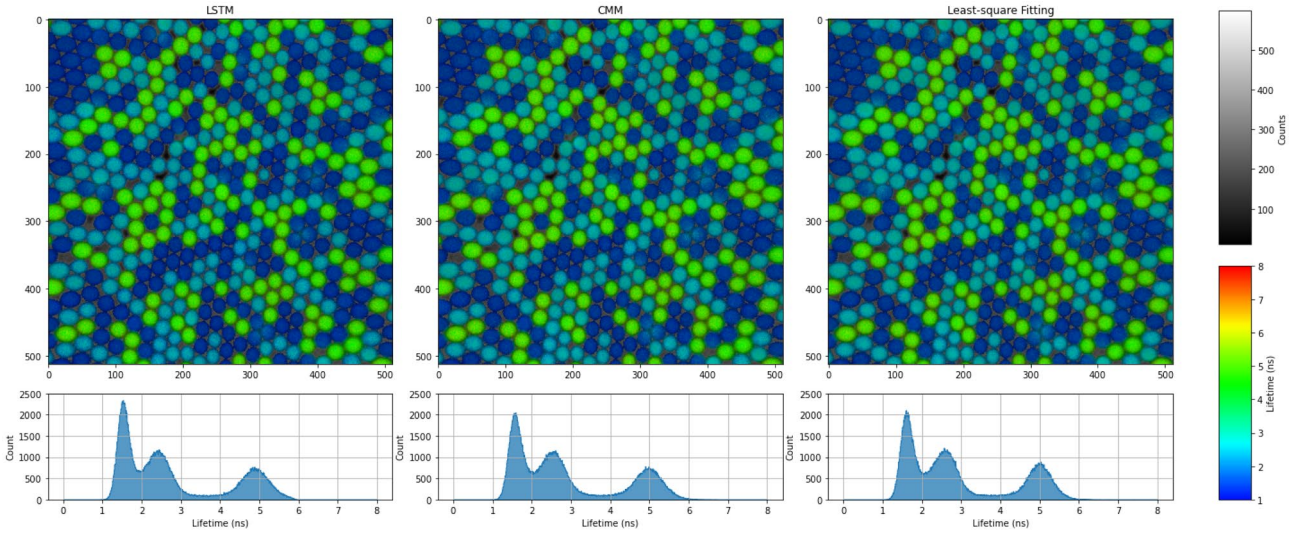


Figure 3. Comparison of LSTM, CMM, and LS Fitting on experimental data. The sample contains a mixture of fluorescent beads with three different lifetimes (1.7, 2.7, and 5.5 ns). The fluorescence lifetime images are displayed using a rainbow scale, where the brightness represents photon counts and the hue represents lifetimes. The lifetime histograms among all pixels are shown below. Most pixels are assumed to contain mono-exponential fluorophores. Two or three lifetimes might be mixed at the edge of the beads.

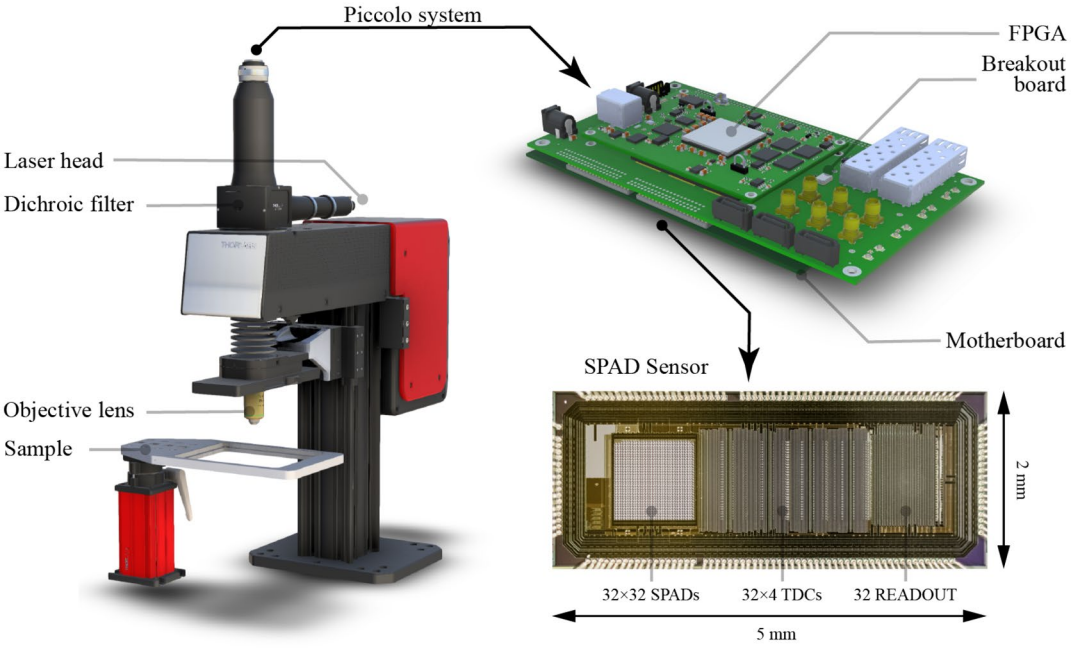


Figure 4. Real-time FLIM system based on the Piccolo 32×32 SPAD sensor and on-FPGA RNNs. The main body of the microscope is from a single-channel Cerna® Confocal Microscope System (ThorLabs, Newton, New Jersey, United States). On the top is the Piccolo system, composed of the SPAD sensor itself, motherboard, breakout board, and FPGA. The SPAD sensor has 32×32 SPADs and 128 on-chip TDCs, offering 50 ps temporal resolution. The FPGA is programmed to control the SPAD sensor and communicate with PC through USB 3. The RNN is also deployed on the same FPGA.

the movement of beads in the images. The result is shown in Fig. 5. The lifetime images are also displayed in rainbow scale. The average photon count for the beads is around 500 per pixel. This illustrates that our system can capture the movement of beads and provide an accurate lifetime estimation. One can also observe that there are some outliers, e.g. dark blue dots and red dots among the green beads. Apart from statistical fluctuations, RNN-based lifetimes tend to be lower when there are not enough photons, which explains why the blue dots are mostly darker than the surrounding pixels.

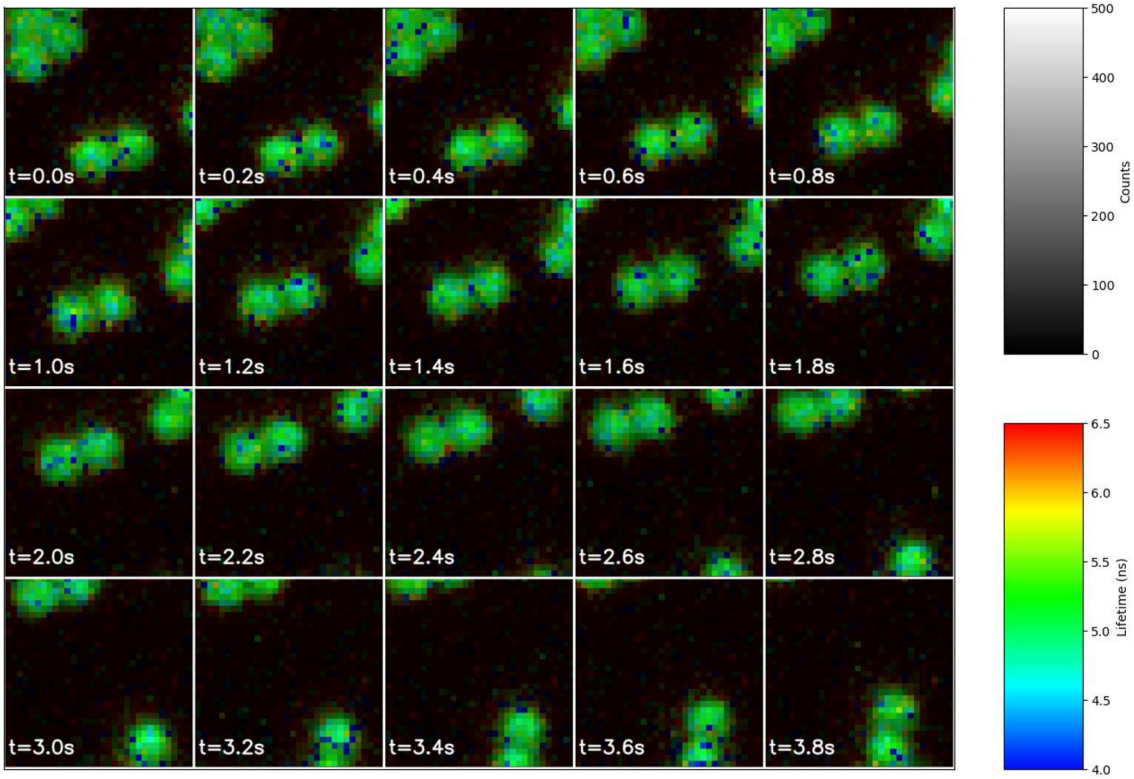


Figure 5. Real-time lifetime image sequence from our FLIM system. The sample contains fluorescent beads with a 5.5 ns reference lifetime. (See the full video in the Supplementary Material).

Discussion

The proposed on-FPGA RNN eliminates the need for histogramming by directly utilizing raw timestamps as input. This transforms the conventional FLI workflow, differentiating it clearly from prevailing NN-based methods. In other words, the paradigm shifts from traditional frame-based processing to event-based processing, responding to each incoming photon. The lifetime is thus estimated on the fly instead of after the acquisition, which reduces the latency and enables real-time lifetime estimation at the edge. The reduction of NN input size, from hundreds of parameters when employing histograms, to a single scalar when operating with timestamps, significantly diminishes the model size. This releases hardware resources when implemented on the FPGA and significantly reduces the burden on data transfer. The quantization of the RNN further reduces the required hardware resources and latency, while the quantization-aware training retains the high accuracy of the RNNs, allowing for multicore parallel processing on the FPGA. Beyond the workflow enhancement, the analysis of synthetic data and CRLB shows that RNNs, especially GRUs and LSTMs, reach excellent accuracy and robustness compared to traditional methods, while retaining higher photon efficiency. Further analysis of experimental data highlights the generalizability of the proposed approach, even when exclusively trained on synthetic datasets. The FLI microscope, integrated with the proposed on-FPGA RNN, successfully demonstrates the anticipated accuracy in lifetime estimation and exhibits real-time processing capabilities.

The adaptability of the proposed approach allows effortless redeployment and migration into other SPAD TCPSC systems. The performance of the whole system can be further improved by using a larger SPAD sensor and by accommodating more RNN cores on the FPGA. A more powerful FPGA or even dedicated neural network accelerator can be used to accommodate more RNN cores. More efficient quantization and approximate schemes such as 8-bit integer quantization can also be explored to reduce resource utilization and latency. In addition, these GRU cores, which are implemented with HLS, can be further optimized by VHDL implementation. In the future, the RNN cores could be implemented on application-specific integrated circuits (ASICs) and stacked together with SPAD arrays by means of 3-D stacking technology, realizing in-sensor processing[44].

Though the proposed system is only used for FLI to this date, it can be easily adapted for other applications by retraining the RNN. It can be further combined with other large-scale neural networks for high-level applications, where the output of the RNN is composed of high-level features learned by neural networks automatically, and it serves as input for other neural networks. The existing FLI-based high-level applications such as margin assessment[5,32] could also be directly incorporated into our system.

Methods

Dataset

The dataset is crucial to the success of neural network models. Curating an experimental dataset for training requires extensive labor, which is time-consuming and costly. The specificity of instruments will bring

bias to the data and thus impede models' generalization. Existing works that use neural networks for fluorescence lifetime estimation all take the approach of training on synthetic datasets and evaluating on experimental datasets[27,28,31,45–47]. The same approach is taken in this work. We construct synthetic datasets for training and evaluation. A small experimental dataset will also be built for the evaluation of our model on real-world data.

Synthetic dataset

A simulation that well captures the features of the real scene is the key to constructing synthetic datasets. To accurately model a real FLI system, we take fluorescence decay, instrument response, background noise, and dark counts into account. The latter two are often neglected in previous studies. Nonetheless, in various scenes, such as fluorescence-assisted surgery, there is a notable presence of significant background noise that cannot be easily dismissed. Minor factors such as differential non-linearity (DNL) of time tagging, pile-up effect, crosstalk among pixels, etc. are omitted in the model. Different from existing NN-based methods, which take histograms as input, we generate synthetic datasets on the timestamp level. Assuming that at most one photon reaches the detector in every repetition period (i.e. the pile-up effect is not considered), the timestamp t, namely the arrival time of photons, is modeled as:

$$t = \sum_{i=1}^{N-1} \mathbf{1}_{k=i}(t_{fluo_i} + t_{irf}) + \mathbf{1}_{k=N} t_{bg}, \tag{1}$$

where $\mathbf{1}$ is the indicator function, k is the component indicator, t_{fluo} is the fluorescence time delay, t_{irf} is the instrument response time delay, t_{bg} is the arrival time of background noise or dark counts, and $N-1$ is the number of the components of fluorescence (the N^{th} component is the background noise). For instance, $N=2$ for the mono-exponential model and $N=3$ for the bi-exponential model.

The component indicator k is a random variable with categorical distribution, representing the source of the incoming photon, which can be either a component of the fluorescence decay or background noise. Hence k is an integer ranging from 1 to N. The probability density function (PDF) of k is

$$f(k|\mathbf{p}) = \prod_{i=1}^{N} p_i^{\mathbf{1}_{k=i}}, \tag{2}$$

where p_i represents the normalized intensity of fluorescence or background noise. When k is given, Eq. (1) can be simplified to

$$t = \begin{cases} t_{fluo_k} + t_{irf}, & \text{if } k \neq N. \\ t_{bg}, & \text{otherwise.} \end{cases} \tag{3}$$

The fluorescence time delay t_{fluo_i} is subject to an exponential distribution. Its PDF is:

$$f(t_{fluo_i}|\tau_i) = \frac{1}{\tau_i} e^{-\frac{t_{fluo_i}}{\tau_i}}, \tag{4}$$

where τ_i is the lifetime of the fluorescence decay.

The instrument response time delay t_{irf} is subject to a Gaussian distribution. Its PDF is:

$$f(t_{irf}|t_0, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t_{irf}-t_0}{\sigma}\right)^2}, \tag{5}$$

where t_0 is the peak position, and σ can be represented by full width at half maximum(FWHM):

$$\sigma = \frac{FWHM}{2\sqrt{2\ln 2}}. \tag{6}$$

The time of arrival of background noise t_{bg} is subject to a uniform distribution. Its PDF is:

$$f(t_{bg}|T) = \frac{1}{T}, \tag{7}$$

where T is the repetition period.

Given a set of the parameters, including N, $p_{\{1,2,\ldots,N\}}$, $\tau_{\{1,2,\ldots,N-1\}}$, t_0, σ, and T, timestamps can be sampled from the above distributions, and thus synthetic datasets can be constructed with different lifetime ranges, background noise ratios, and components of fluorescence. The parameters that determine synthetic datasets are chosen based on common FLI scenarios, which are supposed to cover common photon counts, fluorescence lifetime, IRF, and background noise level. In this work, the FWHM is assumed to be 167.3 ps, in accordance with previous studies[29,30]; t_0 for each sample is generated from a uniform distribution from 0 to 5 ns. To train models in the presence of background noise, p_N for each sample is generated from a uniform distribution from 0 to 10%. Each dataset contains 500,000 samples, and each sample contains 1024 timestamps.

Experimental dataset

Testing the model, which is purely trained on synthetic data, on experimental data is essential to ensure its applicability in real-world scenarios, thus an experimental dataset is curated for evaluation. Fluorescent beads

with reference lifetimes of 1.7, 2.7, and 5.5 ns are adopted as sample (PolyAn GmbH, 11000006, 11010006, and 11020006)[48]. The beads have a 3D-carboxy surface, with a diameter of 6.5 µm. The excitation wavelength is around 488 nm and the emission spectra are 602-800 nm, 545-800 nm, and 559-718 nm, respectively. The fluorescence intensity of these three beads is around 1:2:5. Fluorescent beads with different lifetimes are mixed together with all possible combinations, and put in a 384-well plate for imaging.

A commercial FLIM system, available at the Bioimaging and Optics Platform (BIOP) of EPFL, is utilized to measure the sample and acquire the experimental data. A confocal microscope (Leica SP 8, w/ HyD SMD detector) is used for imaging, a super-continuum laser (NKT Photonics, SuperK Extreme EXW-45) is used for illumination, and a TCSPC module (PicoHarp 300 TCSPC) is used for time-tagging. The sample is excited under a 20 MHz laser, corresponding to a repetition period of 50 ns. The excitation wavelength is 486 nm and the spectrum of the emission filter ranges from 600 to 700 nm. The temporal resolution of time-tagging is 16 ps.

Neural network

The neural network is first built, trained, and evaluated on the PC with PyTorch[49]. Then its weights are quantized and the activation functions are approximated. After that, the neural network is written in C/C++, loading the quantized weights and approximated activation functions, and is further translated into hardware description language (HDL) by Vitis High-level Synthesis (HLS).

Model

Three RNN variants are adopted here, which are simple RNN, GRU, and LSTM. The default models in PyTorch are used, whereas the input size is 1, so selected to accommodate the timestamps. The formula and structure of the simple RNN have been demonstrated in Fig. 1. The formula and structure of GRU and LSTM have been well explained in the literature[40,41]. The numbers of hidden units (hidden size) and layers of RNNs determine the capacity of the model, i.e. the ability to solve complex tasks. Aiming at implementation on the FPGA and even hardware, however, more hidden units and layers will significantly increase the memory consumption and the computation complexity. Considering the hardware limitation, only single-layer RNNs are considered. The numbers of parameters, multiply-accumulate operations, and activations with respect to the hidden size are summarized in Table 3. Besides, a hidden state, a vector as the output of hidden units, has to be stored for every pixel. Therefore, the trade-off between the hidden size and the capability of the RNN has to be considered. After examining the hardware resource on the FPGA, the hidden size is set between 8 to 64. Since the timestamps are processed in real time and are not stored, bidirectional RNNs cannot be used. An FCNN with one hidden layer takes the hidden state as input to predict the lifetime.

Training

Normally, the loss function for RNNs is built on the output of the last timestep or the average output of all timesteps. In fluorescence lifetime estimation, the performance of estimators is supposed to be improved with more photons. Under this principle, we design a weighted mean square percentage error (MSPE) function, assigning more importance to subsequent timesteps:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{N} w_i \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2, \tag{8}$$

where N is the number of timesteps, $\mathbf{y}$ is the ground truth, $\hat{\mathbf{y}}$ the prediction, and w_i the weight at timestep i:

$$w_i = \frac{1}{1 + e^{-\left(\frac{i - N/4}{N/4}\right)}}. \tag{9}$$

The weight function takes a Sigmoid form, assigning weights smaller than 0.5 to estimations based on less than $N/4$ photons, and assigning weights larger than 0.5 to estimations based on less than $N/4$ photons. We assume that at least 256 photons are required to have a relatively accurate estimation, hence $N/4$ is used here while $N = 1024$ in the training set.

The weights for hidden states are initialized by an orthogonal matrix. All biases are initialized with 0s. For LSTM, the weights for cell states are initialized by Xavier initialization[50], and the bias for forget gates is initialized with 1s.

Model	No. parameters	No. MAC Op	No. activation
Simple RNN	$n^2 + 2n$	$n^2 + n$	n
GRU	$3n^2 + 6n$	$3n^2 + 6n$	$3n$
LSTM	$4n^2 + 8n$	$4n^2 + 7n$	$5n$

Table 3. Numbers of parameters, multiply-accumulate (MAC) operations and activations. n is the hidden size. The input size is 1, and the output layer is omitted.

The dataset is randomly split into training, evaluation, and test set, with the ratio of sizes being 8:1:1. The batch size is 32. Adam optimizer is used with an initial learning rate of 0.001[51]. The learning rate decays every 5 epochs at the rate of 0.9. The whole training process takes 100 epochs.

Evaluation
Three metrics are used to evaluate the performance of RNNs and benchmarks on synthetic data, which are:

$$\mathrm{RMSE} = \frac{\sqrt{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}}{N}, \tag{10}$$

$$\mathrm{MAE} = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{N}, \tag{11}$$

$$\mathrm{MAPE} = \frac{\sum_{i=1}^{N}|\frac{y_i - \hat{y}_i}{y_i}|}{N}. \tag{12}$$

Cramer-rao lower bound
Cramer-Rao lower bound (CRLB) gives the best precision that can be achieved in the estimation of fluorescence lifetime[42,52,53]. Mathematically, CRLB expresses a lower bound of variance of estimators and it is proportional to the inverse of the Fisher information:

$$Var(\hat{\theta}) \geq \frac{(f'(x;\theta))^2}{\mathscr{J}(\theta)}, \tag{13}$$

where $f(x;\theta))$ is the PDF and $\mathscr{J}$ is the Fisher Information, which is defined as:

$$\mathscr{J}(\theta) = nE_\theta\left[\left(\frac{\partial}{\partial\theta}\ln f(x;\theta)\right)^2\right]. \tag{14}$$

The CRLB is calculated with open-source software[42].

FPGA implementation
Quantization is an effective way to reduce resource utilization and latency on hardware. In common deep learning frameworks, such as PyTorch or Tensorflow, model weights and activations are represented by 32-bit floating point numbers. However, it would be inefficient to perform operations for floating point numbers with such bitwidth. We aim to quantize the 32-bit floating point numbers with fixed-point numbers and to reduce the bitwidth as much as possible, while maintaining the same model behavior.

Both PyTorch and TensorFlow provide tools of quantization for edge devices, namely PyTorch Quantization and TensorFlow Lite. However, the quantized models rely on their own libraries to run, and the quantized weights cannot be exported. Therefore, we use Python and an open-source fixed point number library to realize a quantized GRU for evaluation. We compare 8-bit, 16-bit, and 32-bit fixed-point numbers to quantize weights and activations separately. The results show that the weights can be quantized to 16-bit fixed point numbers without a significant accuracy drop, and to 8-bit fixed point numbers with an acceptable accuracy drop. Activations can be quantized to 16-bit fixed point numbers without a significant accuracy drop, but 8-bit fixed point quantization will lead the model to collapse. Besides the fixed point precision, we find that the rounding method has a great impact on the performance. Truncating, often the default rounding method, leads to larger errors. Fixed point numbers with convergent rounding have almost the same behavior as floating point numbers. The activation functions are approximated by piecewise linear functions.

The quantized GRU model is then implemented on FPGA. The GRU is written in C++ and compiled to Vivado IP with Vitis HLS. The whole model is divided into two parts: a GRU core and an FCNN. The GRU core is designed to be shared among a group of pixels, and the FCNN will be run sequentially for each pixel after integration. Upon receiving a timestamp, GRU core loads hidden states from block RAMs (BRAMs), updates the hidden states, and sends them back to BRAM. After the integration of each repetition period, the FCNN loads the hidden state from BRAM, and streams the estimated lifetime to a FIFO.

Experimental setup

A real-time FLI microscopy (FLIM) system with our SPAD sensor and on-FPGA RNN is built, which is shown in Fig. 4. The microscope is adapted from a confocal microscope system (Single-Channel Cerna® Confocal Microscope System), though it is only used for widefield imaging in this work. The same fluorescent bead samples are measured, hence a 480 nm pulsed laser (PicoQuant) is utilized. A set of filters is adopted for fluorescence imaging. The excitation filter (Thorlabs FITC Excitation Filter) has a central wavelength of 475 nm with a bandwidth of 35 nm. The emission filter is a long-pass filter (Thorlabs Ø25.0 mm Premium Longpass Filter) with a cut-on wavelength of 600 nm. The dichroic filter (Thorlabs GFP Dichroic Filter) has a reflection band from 452 to 490 nm and a transmission band from 505 nm to 800 nm.

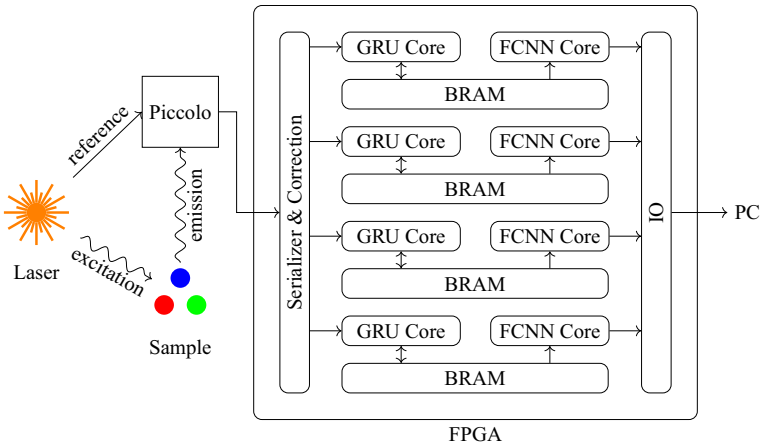


Figure 6. Schematic of the proposed real-time FLIM system with on-FPGA GRUs. A pulsed laser illuminates the sample repeatedly and sends a reference signal to Piccolo to reset TDCs at the same time. The emitted fluorescence photon is then detected by Piccolo and time-tagged, whose arrival time is sent to FPGA in parallel for further processing. The incoming timestamps are serialized and corrected and then sent to GRU cores. The hidden states of GRU are stored in the BRAM. Upon the arrival of a timestamp, the hidden state of the corresponding pixel is read by the GRU core, then the hidden state is updated and written back to the BRAM. After the integration period, the final hidden states are read by FCNN cores and the lifetime is estimated and sent to PC.

The Piccolo system is used for single-photon detection and time tagging[38]. The complete system, along with its components and a micrograph of the Piccolo chip is shown in Fig. 4. Piccolo provides 50-ps temporal resolution and 47.8% peak photon detection probability (PDP). Versions with microlenses are available as well, to improve the light collection efficiency. The median dark count rate (DCR) is 113 cps (per pixel at room temperature). A Xilinx FPGA was used to communicate with the PC and control the sensor. To minimize the system and reduce latency, the RNNs were deployed on the same FPGA.

The schematic of the FPGA design is shown in Fig. 6. Four computation units are realized, each of which is in charge of a quarter of the sensor (32×8 pixels). The timestamps, sent to FPGA in parallel, are serialized and distributed to four computation units based on their SPAD IDs. Each computation unit is composed of one GRU core, one two-layer fully connected neural network (FCNN) core, and one BRAM. The computation speed is mainly limited by the latency of the GRU core, which is 1.05 ns when employing a 160 MHz clock. The photons that arrive when computation units are busy are simply discarded. The four computation units together are capable of processing up to 4 million photons per second.

Data availability

The source code is available at https://github.com/Yang-J-LIN/RnnFlim.

Received: 27 July 2023; Accepted: 25 January 2024
Published online: 08 February 2024

References

1. Datta, R., Heaster, T. M., Sharick, J. T., Gillette, A. A. & Skala, M. C. Fluorescence lifetime imaging microscopy: Fundamentals and advances in instrumentation, analysis, and applications. *J. Biomed. Opt.* **25**, 1–43. https://doi.org/10.1117/1.JBO.25.7.071203 (2020).
2. van Munster, E. B. & Gadella, T. W. J. Fluorescence lifetime imaging microscopy (FLIM). *Microsc. Tech.*https://doi.org/10.1007/b102213 *(2005).*
3. Suhling, K. *et al.* Fluorescence lifetime imaging (FLIM): Basic concepts and some recent developments. *Med. Photonics* **27**, 3–40. https://doi.org/10.1016/j.medpho.2014.12.001 (2015).
4. Wallrabe, H. & Periasamy, A. Imaging protein molecules using FRET and FLIM microscopy. *Curr. Opin. Biotechnol.* **16**, 19–27. https://doi.org/10.1016/j.copbio.2004.12.002 (2005).
5. Unger, J. *et al.* Real-time diagnosis and visualization of tumor margins in excised breast specimens using fluorescence lifetime imaging and machine learning. *Biomed. Opt. Express* **11**, 1216–1230. https://doi.org/10.1364/BOE.381358 (2020).
6. Erkkilä, M. T. *et al.* Macroscopic fluorescence-lifetime imaging of NADH and protoporphyrin IX improves the detection and grading of 5-aminolevulinic acid-stained brain tumors. *Sci. Rep.* **10**, 20492. https://doi.org/10.1038/s41598-020-77268-8 (2020).
7. Weyers, B. W. *et al.* Fluorescence lifetime imaging for intraoperative cancer delineation in transoral robotic surgery. *Transl. Biophotonics*https://doi.org/10.1002/tbio.201900017 *(2019).*
8. Phipps, J., *et al.* Head and neck cancer evaluation via transoral robotic surgery with augmented fluorescence lifetime imaging. In Biophotonics Congress: Biomedical Optics Congress 2018 (Microscopy/Translational/Brain/OTS), CTu2B.3, 10.1364/TRANSLATIONAL.2018.CTu2B.3 (OSA, Washington, D.C.).
9. Becker, W. TCSPC Handbook 9th edn. (2021).
10. Becker, W. Fluorescence lifetime imaging-techniques and applications. *J. Microsc.* **247**, 119–136. https://doi.org/10.1111/j.1365-2818.2012.03618.x (2012).
11. Hirvonen, L. M. & Suhling, K. Wide-field TCSPC: Methods and applications. *Meas. Sci. Technol.* **28**, 012003. https://doi.org/10.1088/1361-6501/28/1/012003 (2017).

12. Kapusta, P., Wahl, M. & Erdmann, R. Advanced photon counting: Applications, methods, instrumentation/volume editors, Peter Kapusta, Michael Wahl, Rainer Erdmann; with contributions by A. Ahlrichs [and fifty others], vol. 15 of Springer Series on Fluorescence, 1617-1306 (Springer, Cham, 2015).
13. Alfonso-Garcia, A. *et al.* Mesoscopic fluorescence lifetime imaging: Fundamental principles, clinical applications and future directions. *J. Biophotonics* **14**, e202000472. https://doi.org/10.1002/jbio.202000472 (2021).
14. Morimoto, K. Megapixel SPAD cameras for time-resolved applications, https://doi.org/10.5075/EPFL-THESIS-8773.
15. Caccia, M., Nardo, L., Santoro, R. & Schaffhauser, D. Silicon photomultipliers and SPAD imagers in biophotonics: Advances and perspectives. *Nucl. Instrum. Methods Phys. Res., Sect. A* **926**, 101–117. https://doi.org/10.1016/j.nima.2018.10.204 (2019).
16. Bruschini, C., Homulle, H., Antolovic, I. M., Burri, S. & Charbon, E. Single-photon avalanche diode imagers in biophotonics: Review and outlook. *Light Sci. Appl.* **8**, 87. https://doi.org/10.1038/s41377-019-0191-5 (2019).
17. Cusini, I. *et al.* Historical perspectives, state of art and research trends of SPAD arrays and their applications (part ii: SPAD arrays). *Front. Phys.* **10**, 606. https://doi.org/10.3389/fphy.2022.906671 (2022).
18. Grinvald, A. & Steinberg, I. Z. On the analysis of fluorescence decay kinetics by the method of least-squares. *Anal. Biochem.* **59**, 583–598. https://doi.org/10.1016/0003-2697(74)90312-1 (1974).
19. Bajzer, E., Therneau, T. M., Sharp, J. C. & Prendergast, F. G. Maximum likelihood method for the analysis of time-resolved fluorescence decay curves. *Eur. Biophys. J.* **20**, 247–262. https://doi.org/10.1007/BF00450560 (1991).
20. Chessel, A., Waharte, F., Salamero, J. & Kervrann, C. A maximum likelihood method for lifetime estimation in photon counting-based fluorescence lifetime imaging microscopy. In 21st European Signal Processing Conference (EUSIPCO 2013), 1–5 (2013).
21. Isenberg, I. & Dyson, R. D. The analysis of fluorescence decay by a method of moments. *Biophys. J .* **9**, 1337–1350. https://doi.org/10.1016/S0006-3495(69)86456-8 (1969).
22. Li, D.-U. *et al.* Real-time fluorescence lifetime imaging system with a 32×32 0.13 μm CMOS low dark-count single-photon avalanche diode array. *Opt. Express*https://doi.org/10.1364/OE.18.010257 *(2010).*
23. Li, D.D.-U. *et al.* Video-rate fluorescence lifetime imaging camera with CMOS single-photon avalanche diode arrays and high-speed imaging algorithm. *J. Biomed. Opt.* **16**, 096012. https://doi.org/10.1117/1.3625288 (2011).
24. Liu, X. *et al.* Fast fluorescence lifetime imaging techniques: A review on challenge and development. *J. Innov. Opt. Health Sci.* **12**, 1930003. https://doi.org/10.1142/S1793545819300039 (2019).
25. Mannam, V., Zhang, Y., Yuan, X., Ravasio, C. & Howard, S. S. Machine learning for faster and smarter fluorescence lifetime imaging microscopy. *J. Phys. Photonics* **2**, 042005. https://doi.org/10.1088/2515-7647/abac1a (2020).
26. Wu, G., Nowotny, T., Zhang, Y., Yu, H.-Q. & Li, D.D.-U. Artificial neural network approaches for fluorescence lifetime imaging techniques. *Opt. Lett.* **41**, 2561–2564. https://doi.org/10.1364/OL.41.002561 (2016).
27. Smith, J. T. *et al.* Fast fit-free analysis of fluorescence lifetime imaging via deep learning. *Proc. Natl. Acad. Sci.* **116**, 24019–24030. https://doi.org/10.1073/pnas.1912707116 (2019).
28. Zickus, V. *et al.* Fluorescence lifetime imaging with a megapixel SPAD camera and neural network lifetime estimation. *Sci. Rep.* **10**, 20986. https://doi.org/10.1038/s41598-020-77737-0 (2020).
29. Xiao, D., Chen, Y. & Li, D.D.-U. One-dimensional deep learning architecture for fast fluorescence lifetime imaging. *IEEE J. Sel. Top. Quantum Electron.* **27**, 1–10. https://doi.org/10.1109/JSTQE.2021.3049349 (2021).
30. Zang, Z., *et al.* Fast fluorescence lifetime imaging analysis via extreme learning machine. arXiv preprint arXiv:2203.13754 10.48550/arXiv.2203.13754 (2022).
31. Chen, Y.-I. *et al.* Generative adversarial network enables rapid and robust fluorescence lifetime image analysis in live cells. *Commun. Biol.* **5**, 18. https://doi.org/10.1038/s42003-021-02938-w (2022).
32. Marsden, M. *et al.* Intraoperative margin assessment in oral and oropharyngeal cancer using label-free fluorescence lifetime imaging and machine learning. *I.E.E.E. Trans. Biomed. Eng.* **68**, 857–868. https://doi.org/10.1109/TBME.2020.3010480 (2021).
33. Sagar, M. A. K. *et al.* Machine learning methods for fluorescence lifetime imaging (FLIM) based label-free detection of microglia. *Front. Neurosci.* **14**, 931. https://doi.org/10.3389/fnins.2020.00931 (2020).
34. Xiao, D. *et al.* Dynamic fluorescence lifetime sensing with CMOS single-photon avalanche diode arrays and deep learning processors. *Biomed. Opt. Express* **12**, 3450–3462. https://doi.org/10.1364/BOE.425663 (2021).
35. Zang, Z. *et al.* Compact and robust deep learning architecture for fluorescence lifetime imaging and FPGA implementation. *Methods Appl. Fluoresc.* **11**, 025002. https://doi.org/10.1088/2050-6120/acc0d9 (2023).
36. Xiao, D., *et al.* Smart Wide-field Fluorescence Lifetime Imaging System with CMOS Single-photon Avalanche Diode Arrays. Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference **2022**, 1887–1890, 10.1109/EMBC48229.2022.9870996 (2022).
37. Elman, J. L. Finding structure in time. *Cogn. Sci.* **14**, 179–211. https://doi.org/10.1207/s15516709cog1402_1 (1990).
38. Zhang, C., Lindner, S., Antolovic, I. M., Wolf, M. & Charbon, E. A CMOS SPAD imager with collision detection and 128 dynamically reallocating TDCs for single-photon counting and 3D time-of-flight imaging. *Sensors* **18**, 4016. https://doi.org/10.3390/s18114016 (2018).
39. Kalyanov, A., Ackermann, M., Russomanno, E., Wolf, M. & Jiang, J. Time-multiplexing approach for fast time-domain near-infrared optical tomography combined with neural-network-enhanced image reconstruction. *In Diffuse Optical Spectroscopy and Imaging IX***12628**, 56–58. https://doi.org/10.1117/12.2670186 (SPIE, 2023).
40. Cho, K., *et al.* Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 10.48550/arXiv.1406.1078 (2014).
41. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735 (1997).
42. Bouchet, D., Krachmalnicoff, V. & Izeddin, I. Cramér-Rao analysis of lifetime estimations in time-resolved fluorescence microscopy. *Opt. Express* **27**, 21239–21252. https://doi.org/10.1364/OE.27.021239 (2019).
43. Lindner, S., *et al.* A novel 32 x 32, 224 Mevents/s time resolved SPAD image sensor for near-infrared optical tomography. Microscopy Histopathology and Analytics JTh5A.6, 10.1364/TRANSLATIONAL.2018.JTh5A.6 (2018).
44. Charbon, E., Bruschini, C. & Lee, M.-J. 3D-stacked CMOS SPAD image sensors: Technology and applications. in *2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 1–4, https://doi.org/10.1109/ICECS.2018.8617983 (IEEE, 12/9/2018 - 12/12/2018).
45. Xiao, D. *et al.* Spatial resolution improved fluorescence lifetime imaging via deep learning. *Opt. Express* **30**, 11479–11494. https://doi.org/10.1364/OE.451215 (2022).
46. Héliot, L. & Leray, A. Simple phasor-based deep neural network for fluorescence lifetime imaging microscopy. *Sci. Rep.* **11**, 23858. https://doi.org/10.1038/s41598-021-03060-x (2021).
47. Yao, R., Ochoa, M., Yan, P. & Intes, X. Net-FLICS: Fast quantitative wide-field fluorescence lifetime imaging with compressed sensing—A deep learning approach. *Light Sci. Appl.* **8**, 26. https://doi.org/10.1038/s41377-019-0138-x (2019).
48. PolyAn GmbH. Fluorescence lifetime beads. https://www.poly-an.de/micro-nanoparticles/other-microparticles/fluorescence-lifetime-applications. Accessed: 2023-10-16.
49. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.*https://doi.org/10.48550/arXiv.1912.01703 *(2019).*
50. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. pp 249–256 (JMLR Workshop and Conference Proceedings, 2010).

51. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 10.48550/arXiv.1412.6980 (2014).
52. Köllner, M. & Wolfrum, J. How many photons are necessary for fluorescence-lifetime measurements?. *Chem. Phys. Lett.* **200**, 199–204. https://doi.org/10.1016/0009-2614(92)87068-Z (1992).
53. Kim, J. & Seok, J. Statistical properties of amplitude and decay parameter estimators for fluorescence lifetime imaging. *Opt. Express* **21**, 6061–6075. https://doi.org/10.1364/OE.21.006061 (2013).

Acknowledgements

The authors acknowledge the financial support provided by US Department of Energy. The authors would also like to extend their gratitude to BIOP, EPFL for their help in the experiments.

Author contributions

Y.L., C.B., and E.C. conceived the experiments, conducted the experiments, and evaluated the results. Y.L. and P.M. prepared the hardware and implemented the firmware for the SPAD sensor. A.A. built the confocal microscope. Y.L. and A.A. prepared the experimental samples. E.C. had the original idea for this work. E.C. and C.B. supervised the project. All authors reviewed the manuscript.

Competing interests

For the sake of transparency, the authors would like to disclose that (i) Edoardo Charbon holds the position of Chief Scientific Officer of Fastree3D, a company making LiDARs for the automotive market, and that (ii) Claudio Bruschini and Edoardo Charbon are co-founders of Pi Imaging Technology. Neither company has been involved with the work or paper.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-52966-9.

Correspondence and requests for materials should be addressed to E.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.